10/830,164 PTO-892

(12) UK Patent Application (19) GB (11) 2 207 264 (13) A

(43) Application published 25 Jan 1989

(21) Application No 8814848

(22) Date of filing 22 Jun 1988

(30) Priority data
(31) 62/155731 (32) 23 Jun 1987 (33) JP
62/239371 24 Sep 1987

(71) Applicant
Mitsubishi Denki Kabushiki Kaisha

(Incorporated in Japan)

2-3 Marunouchi 2-chome, Chiyoda-ku, Tokyo, Japan

(72) Inventors
Shunichiro Nakamura
Harumi Minemura
Tatsuo Minohara

(74) Agent and/or Address for Service
J A Kemp & Co
14 South Square, Gray's Inn, London, WC1R 5EU

(51) INT CL⁴
G06F 15/40

(52) Domestic classification (Edition J):
G4A UB

(56) Documents cited
None

(58) Field of search
G4A
Selected US specifications from IPC sub-class
G06F

(54) Data processing system

(57) A data base horizontal partitioning system is provided for use in a relational data base management system for storing a relation contained in a data base into a plurality of disc storage units 2a-2d by partitioning horizontally the relation on the basis of tuples: wherein when storing the relation having clustered indexes 12 therein into a plurality of disc storage units, and when a physical page as 13a in a disc storage unit is to be filled to a full state with a plurality of tuples in the relation, the page may be divided into two so that one half of thus-divided page may be stored into a disc storage unit which has currently a least number of pages containing the tuples for the relation.

Fig. 1

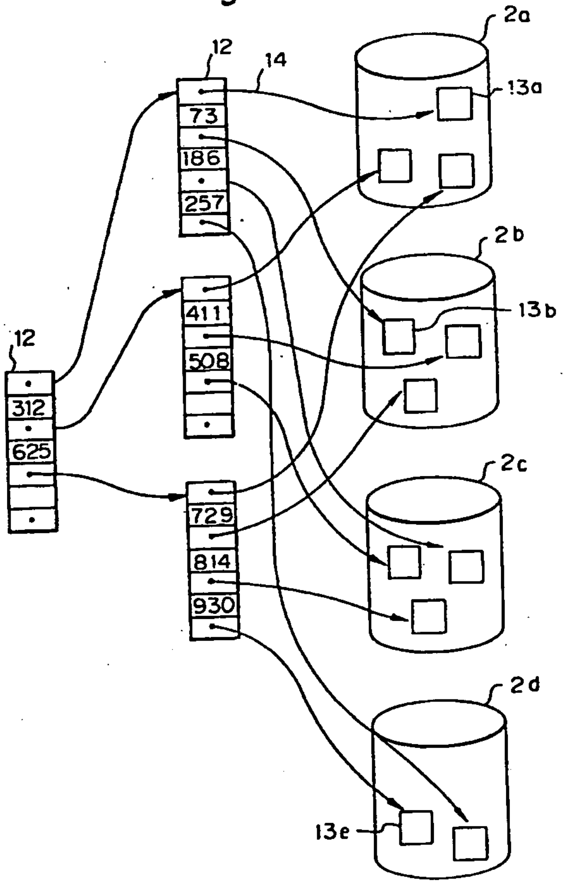


Fig. 1

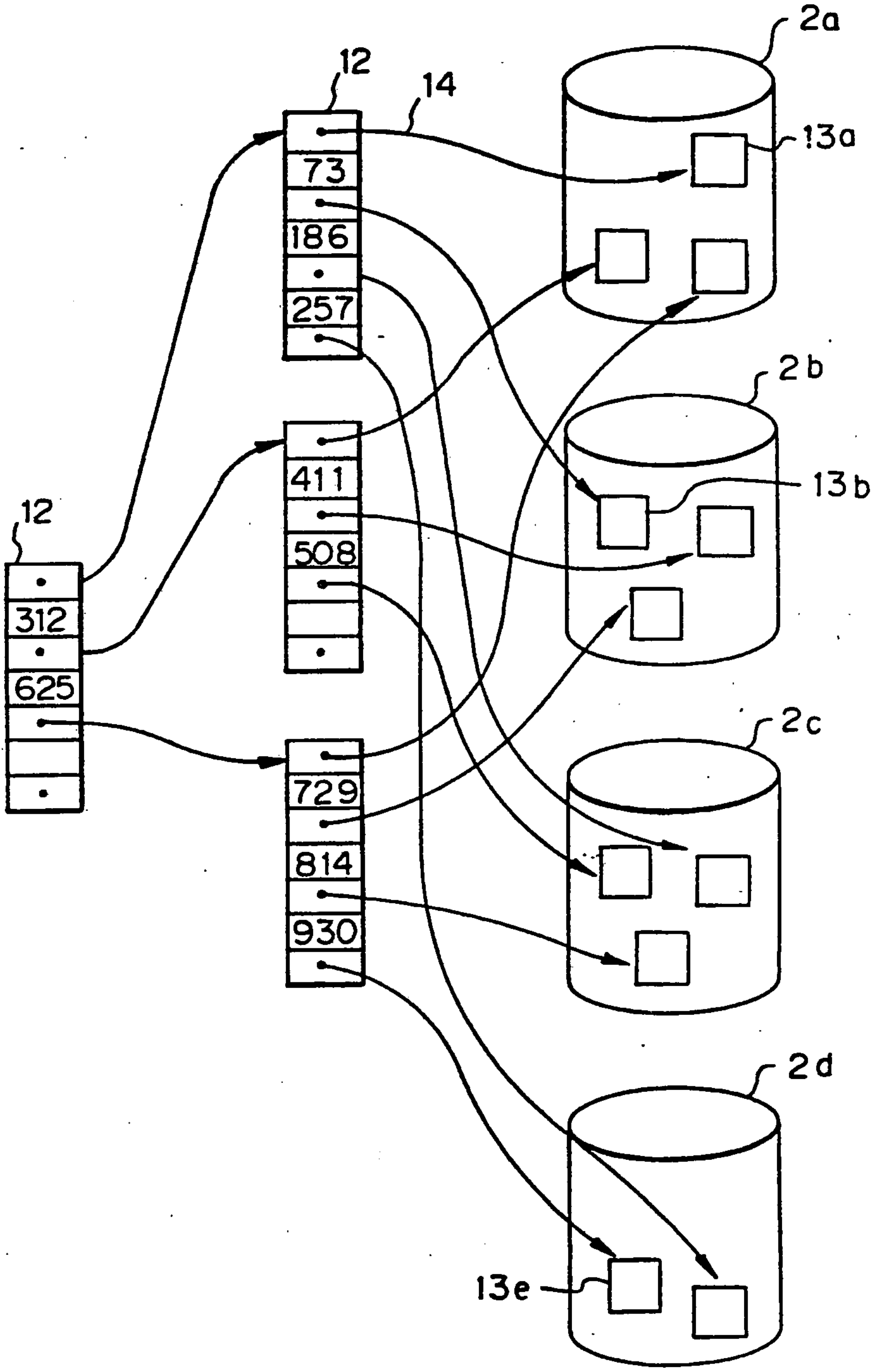
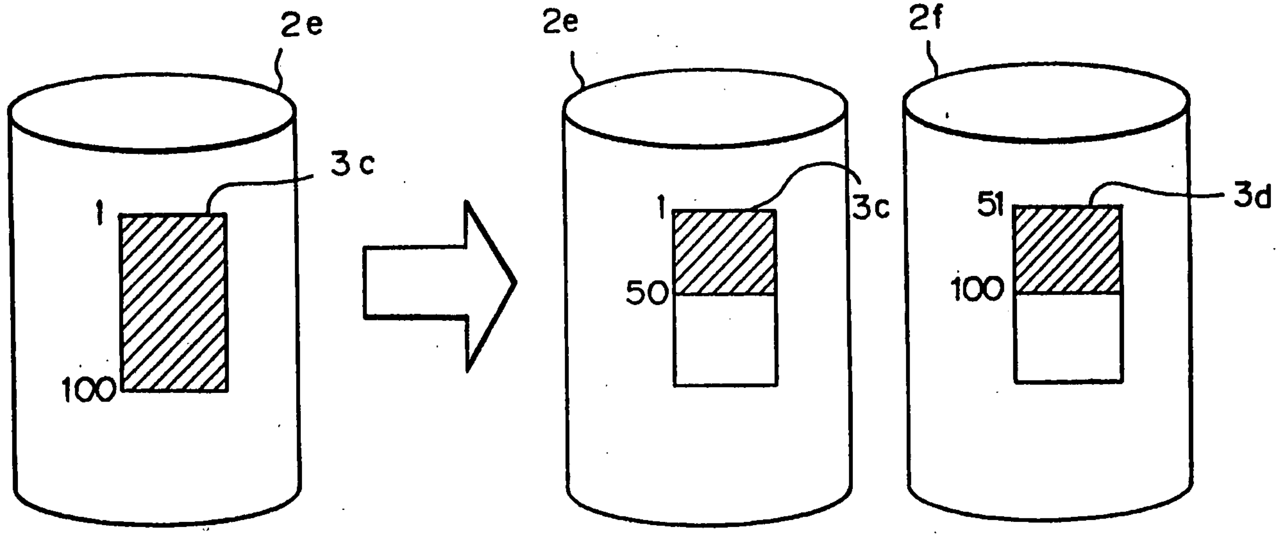

2a
12
14
13a
73
186
257
2b
411
508
13b
12
312
625
2c
729
814
930
2d
13e

Fig. 2



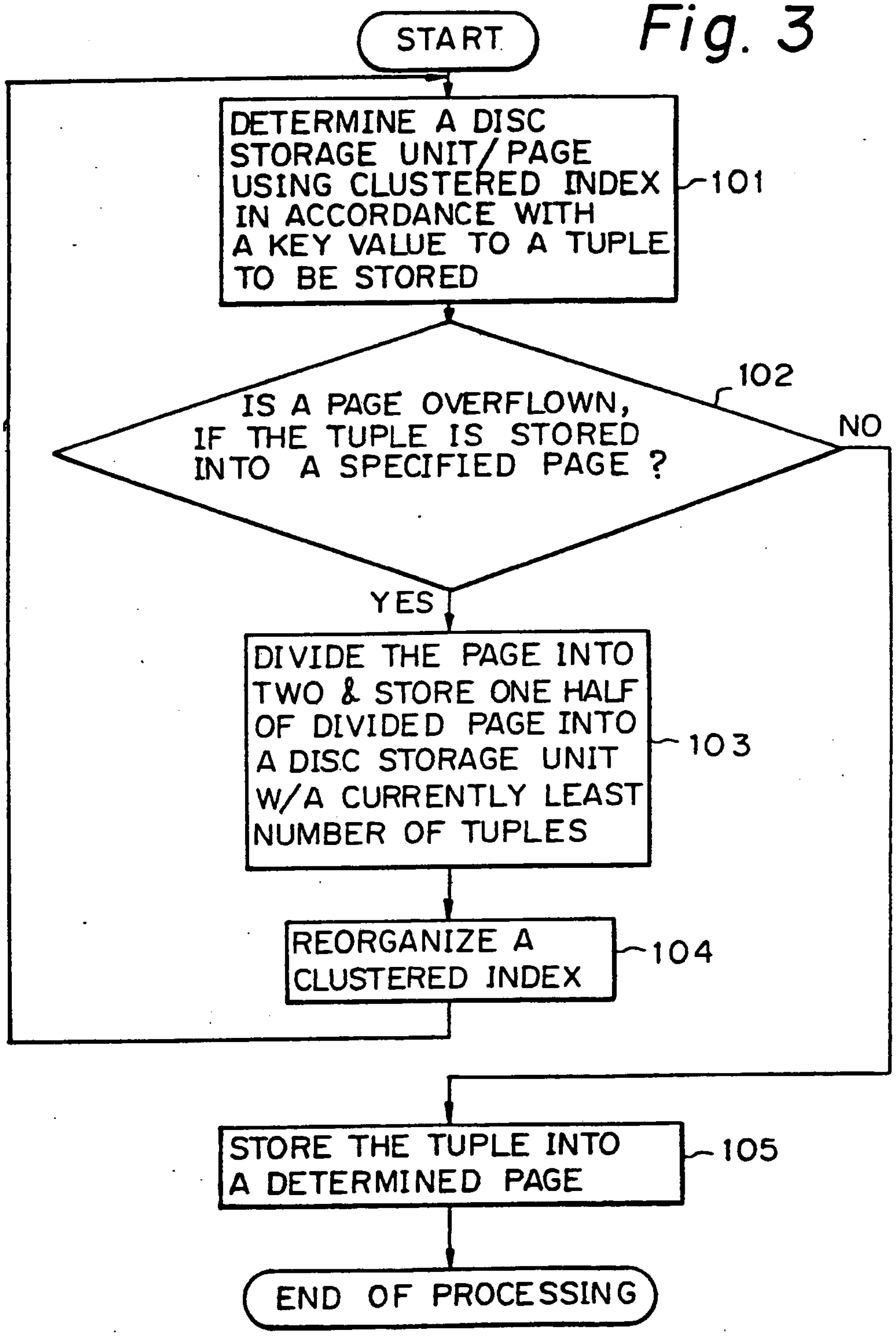

START
DETERMINE A DISC STORAGE UNIT/PAGE USING CLUSTERED INDEX IN ACCORDANCE WITH A KEY VALUE TO A TUPLE TO BE STORED
101
IS A PAGE OVERFLOWN, IF THE TUPLE IS STORED INTO A SPECIFIED PAGE ?
102
NO
YES
DIVIDE THE PAGE INTO TWO & STORE ONE HALF OF DIVIDED PAGE INTO A DISC STORAGE UNIT W/A CURRENTLY LEAST NUMBER OF TUPLES
103
REORGANIZE A CLUSTERED INDEX
104
STORE THE TUPLE INTO A DETERMINED PAGE
105
END OF PROCESSING

Fig. 3

Fig. 4

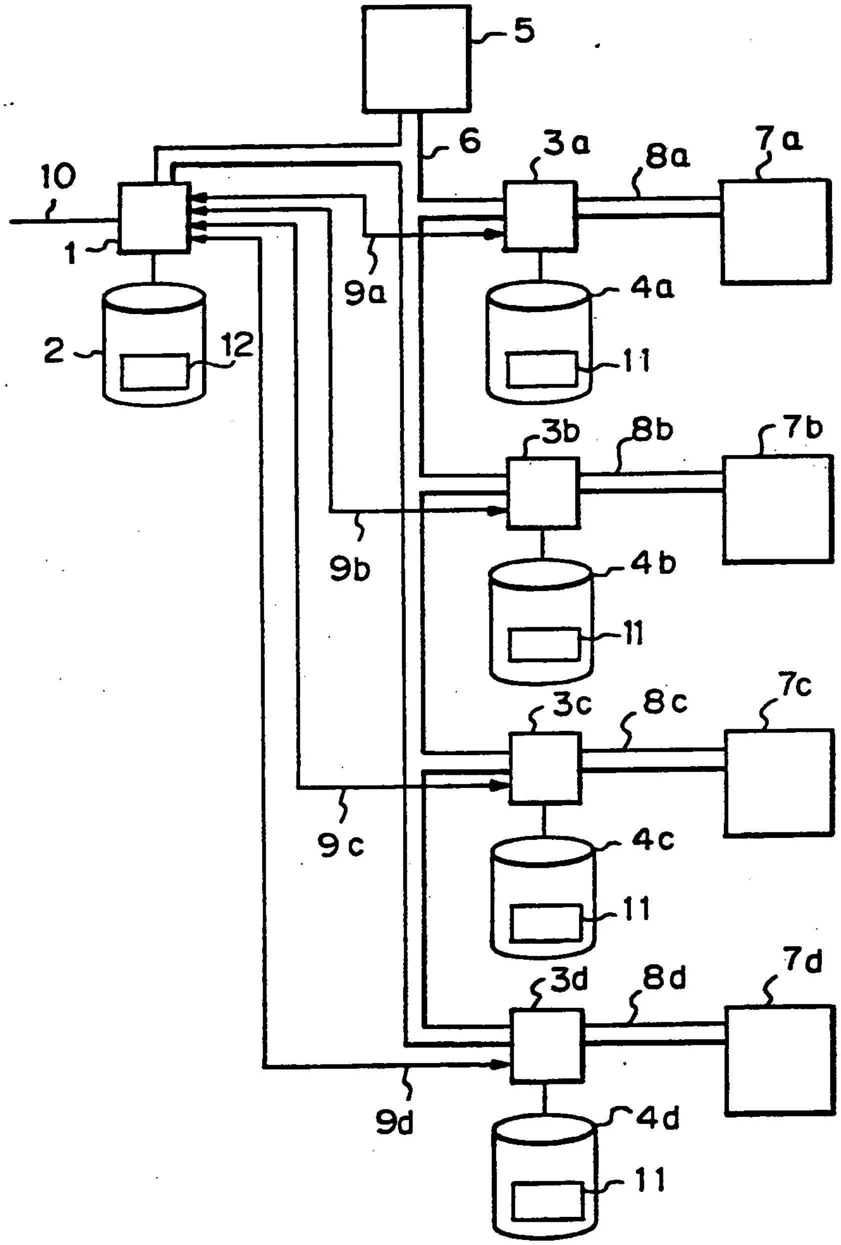

5
6
10
1
2
12
3a
8a
7a
9a
4a
11
3b
8b
7b
9b
4b
11
3c
8c
7c
9c
4c
11
3d
8d
7d
9d
4d
11

Fig. 5

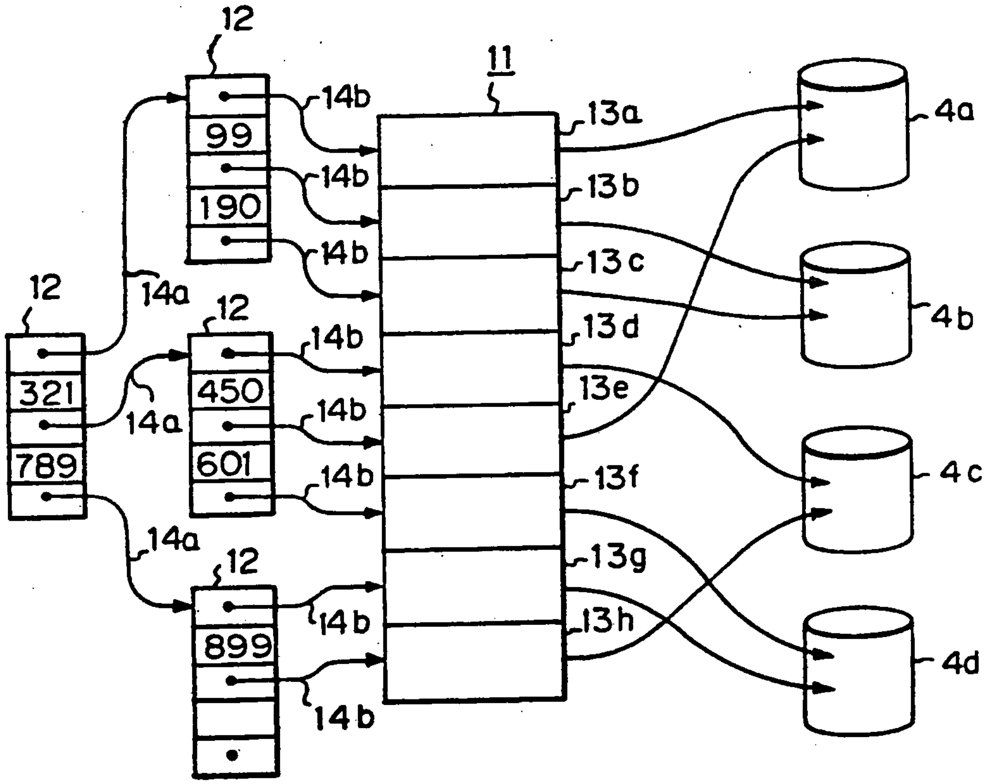

12
11
12
12
12
321
789
99
190
450
601
899
14a
14a
14a
14b
14b
14b
14b
14b
14b
14b
14b
13a
13b
13c
13d
13e
13f
13g
13h
4a
4b
4c
4d

Fig. 6

11 18a 18b 18c 18d

DEPT. NAME	SECTION NAME	NAME OF PERSONNEL	AGES	
GENERAL	GENERAL	山田太郎	40	17a
GENERAL	PERSONNEL	佐藤 弘	35	17b
GENERAL	ACCOUNTANTS'	高橋京子	24	17c
BUSINESS	BUSI. # 2	斉藤 一	32	17d
GENERAL	PERSONNEL	鈴木 明	29	17e
BUSINESS	BUSI. #1	山本 学	27	17f
BUSINESS	BUSI. #2	井上次郎	31	17g
GENERAL	GENERAL	中山花子	29	17h
GENERAL	GENERAL	村上孝志	43	17l
BUSINESS	BUSI. # 1	松本純一	26	17j

Fig. 7

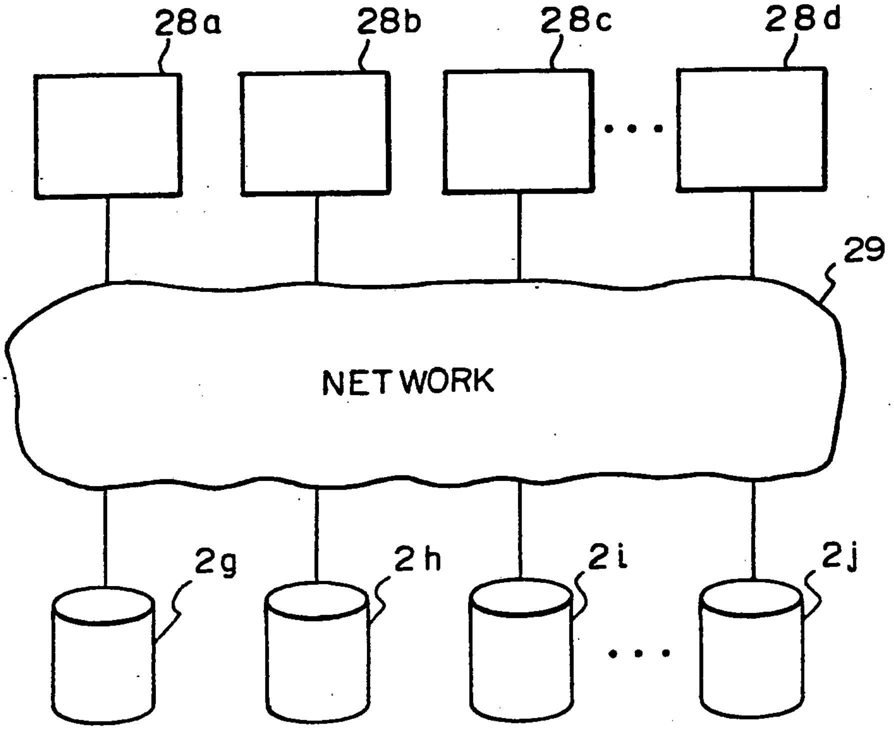


Fig. 8

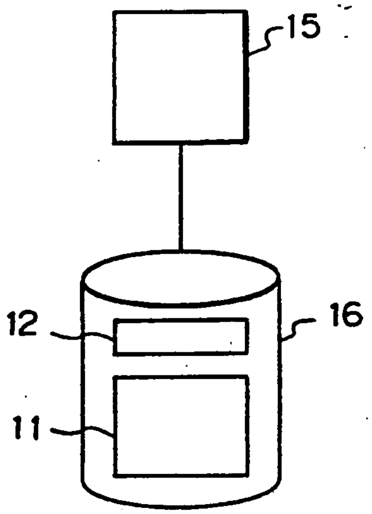
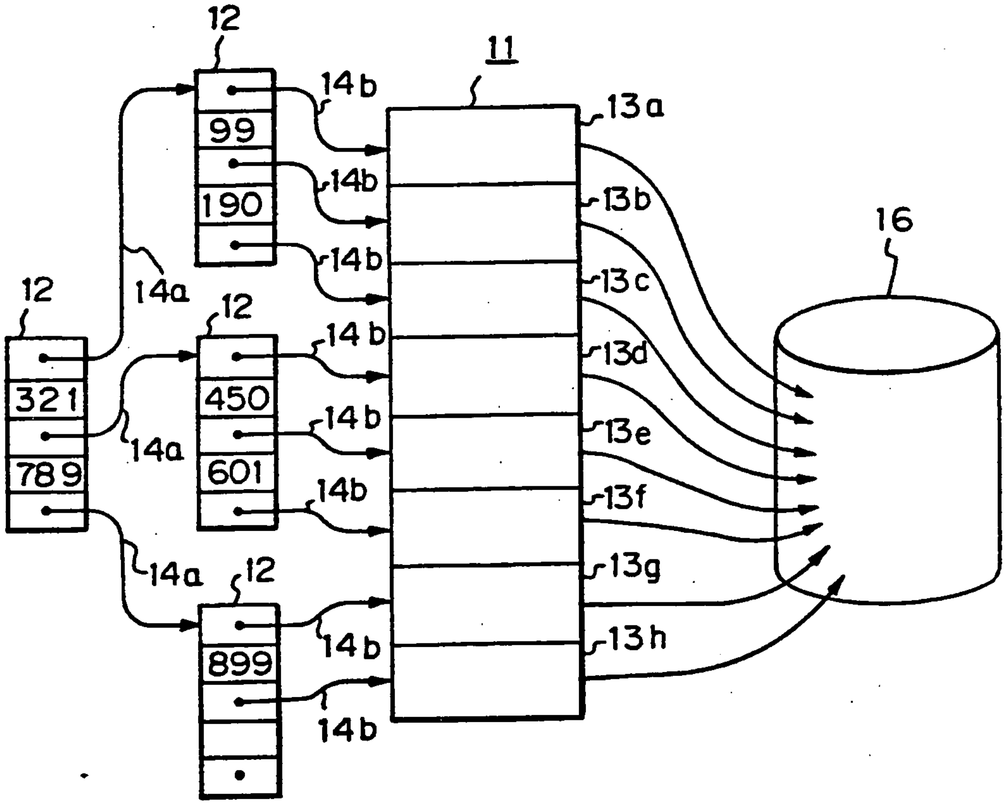


Fig. 9


12
12
12
12
11
14b
14b
14b
14b
14b
14b
14b
14b
14a
14a
14a
99
190
321
789
450
601
899
13a
13b
13c
13d
13e
13f
13g
13h
16

DATA PROCESSING SYSTEM

The present invention relates to a data processing system, and more particularly to a method of storing a data base and to file access processing for use in a relational data base management system.

Before going any further, it may be a help for the reader's better understanding of the invention to give a general review on the known art of relational data base management. In the art of relational data base management system, a data base may be in the form of a collection of tables typically as shown in Fig. 6. An individual table is commonly called a relation 11, each of the items in a table is called an attribute 18, and a record actually loaded with data in an attribute is called a tuple 17.

Referring now to Fig. 7, there is shown the general scheme of a relational data base management system, which is operable in a system wherein a plurality of data processing units 28a - 28d are connected through a network 29 to a plurality of disc storage units 2g - 2j, and wherein one relation is stored in parts distributed amongst the disc storage units 2g - 2j in such a manner that the plurality of data processing units 28a - 28d may in parallel have access to any data contained in the relation as stored in these disc storage units 2g - 2j. In this scheme, one can have a single relation partitioned horizontally into rows forming tuples, this being known as horizontal partitioning.

Thus data as contained in each of rows in a relation is called a tuple. Also, each of items (columns) of a relation is called an attribute. According to the example shown in Fig. 6, a relation is designated by reference number 11, tuples at 17a - 17j, and attributes designated at 18a - 18d, respectively.

With a relation partitioned evenly as noted above, the plurality of data processing units 28a - 28d may read-in data from each of disc storages 2g - 2j in a generally similar time interval, and so it should not happen that one data processing unit will still be reading data, while others have already read data therein during a data handling operation. This allows an increased speed of data processing.

An increase in processing speed in a data base can be achieved by using the system of pointers to ranges of attributes known as "clustered indexes" and described below. However, the typical construction of known horizontal partitioning in a relational data base management system inherently does not provide for clustered indexing and so high speed access to certain tuples in a relation cannot be attained by taking advantage of the use of clustered indexes.

Next, reference is made to Fig. 8 which shows schematically a typical data processing unit. In this figure, an electronic computer or main-frame 15 is connected to a disc storage unit 16. A relation 11 is contained in a data base connected to the disc storage 16, and a clustered index at 12 attached to the relation 11. This cluster index 12 is, for instance, of the type as disclosed in J. D. Ullman's "Principle of Database Systems", paragraph 2.4; issued from Computer Science Press Inc. (Japanese translation: "Database system no genri", translated by Toshiyasu Kunii (phonetic) issued from Nippon Computer Koyokai; p. 71, line 15 through p. 79, line 17). While no particular reference is made to a clustered index in this literature, what is stated as a "B-tree" is, in fact, a clustered index. In general, it is arranged that tuples in a relation 11 are sorted in accordance with a key number particular to a clustered index 12 so as to be stored in a disc storage unit 16. In Fig. 9, there is shown an example wherein clustered indexes are used for attributes (keys) having an integer number ranging from 1 to 1000. The relation 11 may be sorted in accordance with a given key number, partitioned into eight pages 13a - 13h, and stored into the disc storage 16. A pointer 14a is given a number of a page with a pointer 14b stored therein, and the pointer 14b is given page numbers 13a - 13h in the relation 11, respectively. With such arrangement, when a specific key number for a tuple is specified, the page number in which that tuple is stored may immediately be known by referring to the pointers 14a and 14b.

In a relational data base, it is common practice that processing may be directed to a group of tuples in a defined range of certain specified attributes or with a combination of attributes (hereinafter referred to as "keys"). For instance, in the relation 11 shown in Fig. 6, one might wish to obtain an average of the attribute 18d "Ages" of personnel in the "General Section" i.e. with the value "General Section" under the attribute 18b of "Name of Section". In this example, the attribute 18b "Name of Section" would be used as a key.

According to the conventional data processing unit as typically shown in Fig. 8, when processing is performed on a cluster of tuples as defined in terms of their range in connection with the value of key for the clustered index as noted above, the mainframe 15 operates first to refer to the pointers 14a, 14b to the clustered indexes 12, check which page the relevant cluster of tuples are stored, and read them together by page out of the disc storage unit 16 for processing. Since the cluster of tuples as defined in their range in accordance with the key value of the clustered indexes is put together by page and stored into the disc storage unit 16 with their range being physically rather restricted it may suffice to read out only one page from the disc storage unit 16, thus making a processing substantially quicker than the case having no clustered indexes. For instance, in Fig. 9, by virtue of a cluster of tuples existing with the key value in the range from 99 to 190 in the page 13b alone, it would be enough to read the page 13b only from the disc storage unit 16, thus resulting in quicker processing.

While high speed processing may be attained by way of the adoption of the clustered indexes in the conventional data processing unit, as demands for data base management increase, it is difficult to make the data processing even quicker by way of the conventional data processing which is managed by a single main frame per se.

According to the present invention there is provided a data base horizontal partitioning method for use in a relational data base management system for storing a relation contained in a data base into a plurality of storage units by partitioning horizontally the relation on the basis of tuples and storing clustered indexes for said relation wherein when a page in a storage unit would be filled by tuples to be stored, the tuples of that page are divided into two parts and the tuples of one of those parts is stored in a storage unit which currently stores the least number of pages containing the tuples for the relation. The invention also provides a system utilizing the method.

The invention also provides a data processing system including a master processor means, a master storage means connected operatively to said master processor means, a plurality of slave processor means adapted to be controlled by said master processor means, and a plurality of slave storage means connected operatively one to each of said plurality of slave processor means, wherein said master storage means is adapted to store in the form of B-tree structure a clustered index for either an attribute or a combination of attributes of a relation to be processed in a relational data base and said plurality of slave storage means are adapted to store dispersed between them a relation in the data base which is partitioned into pages whereby said plurality of slave processor means may execute in parallel a plurality of processings on a cluster of tuples defined by range in connection with a given key value of said clustered index.

The invention also provides a corresponding method of data processing.

The invention further provides a relational data base processing system in which data in a relation is horizontally portioned into a number of pages and in which an index is maintained in terms of pointers to ranges of an index key to the relation, wherein tuples of said relation are stored dispersed between a plurality of storage means and wherein where a page would be filled by a group of tuples to be stored therein, a number of the tuples of the group on that page are transferred to the storage means which currently holds the least number of pages of the relation.

The invention also provides a corresponding method of data base processing.

Thus with the present invention a relational data base management system may have a desired horizontal partitioning of a data base enabling an even partitioning of a relation having clustered indexes. This can afford a desired quicker processing on a cluster of tuples as defined by range in connection with the key values of clustered indexes in a relational data base.

Preferably, when storing a relation with clustered indexes comprised of a B-tree structure into a plurality of disc storage units, and when a physical page in the disc storage unit is to be filled with a plurality of tuples in the relation, the tuples in the page may be divided into two parts in such a manner that one of the parts may be stored into the disc storage unit which currently has the smallest number of pages containing the tuples for the relation, thus effecting an even horizontal partitioning of a data base.

Also, with this arrangement of data processing unit according to the invention, clustered indexes in a relation may be stored in a primary storage means for a primary processing system, a relation partitioned on a by-page basis may be stored dispersed into secondary storage means operatively connected to a plurality of secondary processing units, and processing on a cluster of tuples defined by range in connection with key values of clustered indexes may be performed in parallel and at a high rate by the respective secondary processing units.

Thus by virtue of the arrangement of a cluster of tuples defined by range in connection with key values of clustered indexes being stored grouped on a by-page basis in secondary e.g. disc storages connected to a plurality of secondary processing units, processing on such a cluster of tuples may be performed in parallel, to effect a high speed data processing.

As outlined hereinbefore, with the advantageous arrangement according to the invention that a relation even with clustered indexes may be partitioned horizontally and evenly, in a system permitting a plurality of data processing units to make concurrent access to a plurality of storage means e.g. discs, there is attainable, in addition to high speed processing by taking advantage of clustered indexes, the effect such that each data processing unit may read-in data in a similar time interval for the processing of a total record search, thereby to allow data reading in a minimum time interval.

A further advantage of the invention is that processing on a cluster of tuples defined by range in connection with key values of clustered indexes may be performed in parallel by a plurality of secondary processing units, and thus that processing on a cluster of tuples so defined may be executed in parallel and at a high rate.

The invention will be further described by way of non-limitative example with reference to the accompanying drawings, in which:

Fig. 1 is an explanatory schematic view showing the general status of partitioning of a relation having clustered indexes by way of a preferred embodiment of the present invention;

Fig. 2 is an explanatory schematic view showing an example of dividing a page in a relation by way of a preferred embodiment of the present invention;

Fig. 3 is a flow chart showing a program in a sequence to practice by way of a preferred embodiment of the present invention;

Fig. 4 is a block diagram showing a data processing unit by way of a preferred embodiment of the present invention;

Fig. 5 is a schematic view showing a status of partitioning of a relation having clustered indexes by way of a preferred embodiment of the present invention;

Fig. 6 is an explanatory schematic diagram showing an example of a typical relation in a relational data base;

Fig. 7 is a schematic diagram showing an example of a system construction of the invention;

Fig. 8 is a block diagram showing the general construction of a typical conventional data processing unit; and

Fig. 9 is a schematic diagram showing a typical arrangement of storing a relation having clustered indexes in the conventional data processing system.

The preѕent invention will now be explained in detail by way of a preferred embodiment thereof in conjunction with accompanying drawings herewith. Referring first to Fig. 1, there is shown a typical clustered index 12.

Each of the tuples is seen stored in sorted order in pages 13a - 13e which are of a physical storing unit in a disc storage units 2a - 2d. In Fig. 1, there is shown a typical example of storage wherein clustered indexes are given to attributes having an integer ranging from 1 to 1000, wherein a cluster of tuples having a lower key value than 73 are, for instance, stored in a page 13a of a disc storage 2a, while those tuples having key values ranging from 73 through 186 are then stored in a page 13b of a disc storage 2b. In this manner, when specifying a key value of a tuple, a particular disc storage and a specific page may be accordingly determined by a pointer 14 as belonging to a certain clustered index.

When storing a specific tuple in a relation having clustered indexes into a disc storage unit, as typically shown in Fig. 3 flow chart, a particular clustered index may be referred to in accordance with a key value of corresponding tuple to determine a disc storage and a page to be stored (Step 101), and when having that tuple stored into the thus-determined page, a determination is made as to whether the specific page is filled up or not (Step 102). If it is determined as a result of this examination that the page would overflow, this page may be divided into two. The tuples of one of such divided half page is to be transferred to a disc storage currently having the least number of tuples stored (Step 103). Fig. 2 shows this situation, wherein it is shown that when tuples with an integral key value ranging from 1 to 100 are stored in a page 3c of the disc storage unit 2e, and when tuples with this range of key value from 1 to 100 are to be stored into this page 3c, this page is divided into two as it is filled up, and that tuples with a key value ranging from 1 to 50 are stored in the original page 3c, while tuples with a key value ranging from 51 to 100 are stored into the other page 3d, respectively. It is to be noted that it is a specific disc storage 2f that is selected for storing the page 3d and that has a currently least number of pages comprised of tuples in a relation to which these tuples belong. Upon the dividing of the page, the following step is reorganization of the clustered indexes (Step 104), and subsequently, a series of Steps 101 et. seq. are followed in repetition. If no overflow of the page is found in Step 102, these tuples may be stored into the page determined in Step 105.

Referring to this embodiment, while an explanation was given on the system structure such that there is provided a network 29 intercommunicating the disc storage units 2g - 2j and the data processing units 28a - 28d, any type of network may of course be feasible in practice, such as a ring type, a single-bus type or the like, which may well serve to an equal effect with that of the embodiment noted above.

Fig. 4 is a block diagram showing a preferred embodiment of a data processing system according to the invention. In Fig. 4, there are shown a primary or master processing unit designated at 1 (hereinafter referred to as "master processor"), a primary or master disc storage unit at 2 connected to the master processor (hereinafter referred to as "master disc unit"), a series of secondary or slave processing units at 3a - 3d (hereinafter referred to as "slave processors"), and a series of secondary or slave disc storage units at 4a - 4d connected to the slave processors (hereinafter referred to as "slave disc units"). Also shown are a common memory at 5, which may be accessed by the master processor 1 and the slave processors 3a - 3d by way of a common bus 6, a plurality of local memories at 7a - 7d, which may be accessed by the slave processors 3a - 3d by way of local buses 8a - 8d, a plurality of interrupt signal lines at 9a - 9d used for a communication between slave processors 3a - 3d by way of an interrupt signal, and an input/output line at 10 for input and output of data between another computer or an external terminal and the master processor. Also shown is a relation designated at 11, which may be divided into a prefixed (e.g., 2 K bytes) unit (hereinafter referred to as a "page") and may be stored divisionally on the basis of this page into the slave disc units 4a - 4d. There are also seen a series of clustered indexes designated at 12 for the relation 11, which is stored into the master disc unit 2. The clustered indexes 12 are of an index comprised of a B-tree structure, and the tuples in the relation 11 may be sorted in accordance with a given key value to the clustered indexes 12, and may be divided into a desired number of pages to be stored into the slave disc units 4a - 4d. Fig. 5 is a schematic diagram showing by way of an example the provision of clustered indexes on the attributes (keys) having a range of integer numbers from 1 to 1000. Relation 11 may be sorted in accordance with a given key value, and may be partitioned into eight pages 13a - 13h. Pages 13a - 13e may be stored in the slave disc unit 4a, pages 13b and 13c stored in the slave disc unit 4b, pages 13d and 13h stored in the slave disc unit 4c, pages 13f and 13g are stored in the slave disc unit 4d, respectively. For example, tuples with a key value smaller than 99 may be stored in page 13a of the slave disc unit 4a, and those with a key value ranging from 99 to 190 stored in page 13b of the slave disc unit 4b, respectively. A pointer 14a is provided with the number of a page in the master disc unit in which a pointer 14b is stored, and a pointer 14b is provided with the number of a slave disc unit in which the relation 11 is stored and the number of a page of this disc unit. With this arrangement, when a key value of tuple is specified, a specific slave disc unit and page in which this specific tuple is stored may be known by routing a pointers 14a and 14b to the clustered index 12, accordingly.

Referring now to the embodiment shown in Fig. 4, the operation how to process on a cluster of tuples as defined in their range in connection with a given key value of the clustered indexes will be explained. In this system, a cluster of tuples as such defined in their range is divided into a plurality of pages and stored possibly evenly in division among four of the slave disc units. For example, according to the example shown in Fig. 5, it is noted that a cluster of tuples having key values ranging from 190 through 789 is divided into four pages 13c through 13f, and stored in the slave disc units 4b, 4c, 4a and 4d, respectively. When a demand for processing on the cluster of tuples is made from another computer or an external terminal by way of the input/output line 10, the master processor 1 operates to refer to the pointers 14a, 14b to the clustered indexes 12, seek numbers of a slave disc unit and of a page in which the relevant cluster of tuples are stored divisionally, write into the common memory 5 a command comprising a content of processing, a page number, etc. to each of the slave processors 3a through 3d, interrupt in succession each of the slave processors 3a - 3d by way of interrupt signal lines 9a - 9d, and inform each of the slave processors 3a - 3d of the existence of a processing to be executed. Each of these slave processors 3a - 3d operates then to read a command directed to itself from the common memory 5, refer to a page number contained in this command, and read out a relevant page from each of the slave disc units 4a - 4d. This readout and processing of a page from these slave disc units 4a - 4d may be done in parallel on the part of each of the slave processors 3a - 3d. Upon completion of such processing, each of such slave processors 3a - 3d operates to write the results of processing into the common memory 5, and interrupt the master processor 1 by way of the interrupt signal lines 9a - 9d to report the completion of processing. Upon an interrupt from all of the slave processors 3a - 3d, the master processor 1 operates then to read out all results of processing from the common memory 5, and return such results to another computer or an external terminal by way of the input/output line 10.

It is now clear that the objects as set forth hereinbefore among those made apparent from the preceding description are efficiently attained, and while the numbers of such components as the slave processor 3a - 3d, the slave disc units 4a - 4d, the local memories 7a - 7d, the local buses 8a - 8d, and the interrupt signal lines 9a - 9d are four according to the preferred embodiment of the invention as noted above, it is to be understood that they may of course be any numbers such as one or more, respectively.

While there are provided one each of the master disc unit 2 and the slave disc units 4a - 4d in the master processor 1 and in each of the slave processors 3a - 3d, respectively, according to the embodiment noted above, it is also to be understood that they may naturally be of any numbers of two or more.

The term B-tree used in the description and claims is not intended to be limited to a binary tree but to include ternary or higher order trees.

CLAIMS

1. A data base horizontal partitioning method for use in a relational data base management system for storing a relation contained in a data base into a plurality of storage units by partitioning horizontally the relation on the basis of tuples and storing clustered indexes for said relation wherein when a page in a storage unit would be filled by tuples to be stored, the tuples of that page are divided into two parts and the tuples of one of those parts is stored in a storage unit which currently stores the least number of pages containing the tuples for the relation.

2. A method according to claim 1 wherein the storage units are accessible in parallel.

3. A data processing system including a master processor means, a master storage means connected operatively to said master processor means, a plurality of slave processor means adapted to be controlled by said master processor means, and a plurality of slave storage means connected operatively one to each of said plurality of slave processor means, wherein said master storage means is adapted to store in the form of B-tree structure a clustered index for either an attribute or a combination of attributes of a relation to be processed in a relational data base and said plurality of slave storage means are adapted to store dispersed between them a relation in the data base which is partitioned into pages whereby said plurality of slave

processor means may execute in parallel a plurality of processings on a cluster of tuples defined by range in connection with a given key value of said clustered index.

4. A system according to claim 1, 2 or 3 wherein said storage means comprise disc storage means.

5. A relational data base processing system in which data in a relation is horizontally portioned into a number of pages and in which an index is maintained in terms of pointers to ranges of an index key to the relation, wherein tuples of said relation are stored dispersed between a plurality of storage means and wherein where a page would be filled by a group of tuples to be stored therein, a number of the tuples of the group on that page are transferred to the storage means which currently holds the least number of pages of the relation.

6. A data base horizontal partitioning system for a relational data base management system in which a relation contained in a data base is stored in a plurality of storage units by partitioning horizontally the relation on the basis of tuples and in which clustered indexes for said relation are stored, wherein when a page in a storage unit would be filled by tuples to be stored, the tuples of that page are divided into two parts and the tuples of one of those parts is stored in a storage unit which currently stores the least number of pages containing the tuples for the relation.

7. A system according to claim 6 wherein the

storage units are accessible in parallel.

8. A data processing method for use on a system including a master processor means, a master storage means connected operatively to said master processor means, a plurality of slave processor means adapted to be controlled by said master processor means, and a plurality of slave storage means connected operatively one to each of said plurality of slave processor means, wherein a clustered index for either an attribute or a combination of attributes of a relation to be processed in a relational data base is stored on said master storage means in the form of B-tree structure and a relation in the data base is partitioned into pages and stored dispersed between said plurality of slave storage means whereby a plurality of processings on a cluster of tuples defined by range in connection with a given key value of said clustered index may be executed in parallel by said slave processing means.

9. A relational data base processing method in which data in a relation is horizontally portioned into a number of pages and in which an index is maintained in terms of pointers to ranges of an index key to the relation, wherein tuples of said relation are stored dispersed between a plurality of storage means and wherein where a page would be filled by a group of tuples to be stored therein, a number of tuples of the group on that page are transferred to the storage means which currently holds the least number

of pages of the relation.

10. A data base processing system constructed and arranged to operate substantially as hereinbefore described with reference to and as illustrated in the accompanying drawings.

11. A data base processing method substantially as hereinbefore described with reference to and as illustrated in the accompanying drawings.

Published 1988 at The Patent Office, State House, 66/71 High Holborn, London WC1R 4TP. Further copies may be obtained from The Patent Office,
Sales Branch, St Mary Cray, Orpington, Kent BR5 3RD. Printed by Multiplex techniques ltd, St Mary Cray, Kent. Con. 1/87